

## 面向地基激光点云的树木枝叶分离

卢华清\*, 伍吉仓, 张子健

同济大学测绘与地理信息学院, 上海 200092

**摘要** 地面激光扫描技术为树木三维信息的获取提供了一种高效、准确的手段。针对目前树木点云枝叶分离方法复杂、自动化程度较低、效果不理想等问题,提出了一种结合最短路径分析与图分割算法的树木点云枝叶分离方法。该方法首先利用最短路径分析算法构建了枝干骨架,在此基础上选择骨架邻近点提取枝干点云,然后利用图分割算法补充主要枝干中漏分的点,实现高精度的枝叶分离。以开源数据中平均点间距不同的树木点云为例,该方法对三类数据的分类精度分别达到 0.9697、0.9469 和 0.9314, kappa 系数均在 0.84 以上,可以有效解决树木细小枝干提取不完整以及枝干漏分离等问题。研究表明,所提方法能够有效分离树木点云中的树叶点和枝干点,且鲁棒性较高,为后续的枝干结构特征提取提供了技术支持。

**关键词** 遥感; 地基激光雷达; 枝叶分离; 最短路径分析; 图分割; 逐点特征

**中图分类号** TN958.98

**文献标志码** A

**DOI:** 10.3788/CJL202249.2310001

## 1 引言

森林树木的结构,如树干形态、分枝模式和叶片的空间分布,直接影响树木的光合作用和蒸散过程,最终影响森林的碳储量和储水量<sup>[1]</sup>。然而,通过手工度量来精确量化树结构是困难且耗时的<sup>[2]</sup>。

地面激光扫描(TLS)能够快速、自动地记录树木结构的毫米级细节并提供物体的辐射特性,因此在植被研究中具有很大的应用潜力<sup>[3]</sup>。在获取 TLS 树木点云后,量化不同物种之间树木结构的差异,实现树木中树叶与枝干结构精确的分离,对于理解树木结构与树木生理功能之间的关系非常重要<sup>[4]</sup>。例如,开展生物多样性保护和气候适应的研究需要了解树木中的树叶分布,因为它会影响光合作用和蒸散过程<sup>[5]</sup>。监测森林生态系统中树木的碳储量和地上生物量(AGB)需要精确量化木材体积的空间分布<sup>[6-7]</sup>。这些研究对象往往是单一的树叶或枝干结构<sup>[8]</sup>,而其他树木结构的存在将会对研究结果造成较大影响。Li 等<sup>[9]</sup>在研究树木辐射状态和光合作用过程与冠层树叶面积及其空间分布的关系时,发现木材的存在会人为地增加表观树叶含量。Calders 等<sup>[10]</sup>基于树木的木材结构,使用定量结构模型(QSM)来估计树木木材体积和 AGB, QSM 可以重建单个树木的三维枝干结构。但精确的 QSM 重建依赖于过滤良好的单树枝干点云,而树叶的存在会使 AGB 估计量偏高。因此,在获取

TLS 树木点云后需要进行正确分割和分类<sup>[11]</sup>,完成高精度的枝叶分离。

目前,国内外许多学者提出了多种树木点云枝叶分离方法。Béland 等<sup>[12]</sup>利用树叶点和枝干点在激光扫描仪的工作波长下存在强度值差异这一特性确定分离阈值,将树叶和枝干分离。但该方法需要对激光雷达仪器进行精确校准。Tao 等<sup>[13]</sup>对树木点云进行水平切片,之后基于几何基元对每个切片进行检测以识别和提取一阶、二阶和三阶树木分支,仅利用  $x$ 、 $y$  和  $z$  坐标就可从原始 TLS 点云中提取枝干点。但该方法通常计算量大且耗时,尤其是在 TLS 点云数量极其庞大的情况下,且对稀疏树木点云的分离结果不理想。Ma 等<sup>[14]</sup>针对点云中的每个点计算一组特征,并使用高斯混合模型(GMM)对它们进行分类。该方法使用监督分类算法,可以训练任意树木结构,但需要大量的人工干预以选择训练数据,且对于不同环境下的树木或不同树种均需要重新训练数据,并不具有通用性。Ferrara 等<sup>[15]</sup>利用枝干点和树叶点体素密度不同的特性,提出了一种基于密度的噪声空间聚类(DBSCAN)算法来自动分离非落叶乔木枝叶。这种方法需要将 TLS 数据体素化,与原始 TLS 数据相比,需要消耗更多的内存空间,且无法有效分离出细小枝干。

针对目前树木 TLS 点云枝叶分离存在的分离结果差、分离效率低、自动化程度低等问题,本文对基于几何特征的无监督分类方法进行进一步优化,提

收稿日期: 2021-12-28; 修回日期: 2022-01-30; 录用日期: 2022-03-24

基金项目: 国家自然科学基金(42074022)

通信作者: \*2033690@tongji.edu.cn

出一种结合最短路径分析算法和图分割算法的枝叶分离算法,该方法不仅可以实现更高精度的 TLS 树木点云枝叶分离,且适用于不同点密度数据,鲁棒性高,为树木枝干模型重建、叶面积指数提取等提供了支持。

## 2 枝叶分离方法

本文所提的枝叶分离算法利用几何特征和结构分析对单棵树的点云进行分类。该算法由三个主要步骤构成:首先是采用基于点几何特征和点云密度的图分

割算法。该算法根据点云中点  $i$  的三维坐标向量  $P_i(x_i, y_i, z_i)$  构建其邻域的协方差矩阵,并根据此矩阵计算出相应点的三个特征值  $\lambda_1, \lambda_2$  和  $\lambda_3$  及相应的特征向量  $e_1, e_2$  和  $e_3$ ,最终结合局部点云密度进行树木枝叶粗分离。其次是最短路径检测,利用 Dijkstra 算法<sup>[16]</sup>计算出树木点云中最低点到其余点的最短路径,之后根据树木生长结构提取出枝干骨架,并基于枝干骨架提取枝干点以实现树木枝叶粗分离。最后,结合两次粗分离结果,实现最终的树木枝叶精分离。具体的算法流程如图 1 所示。

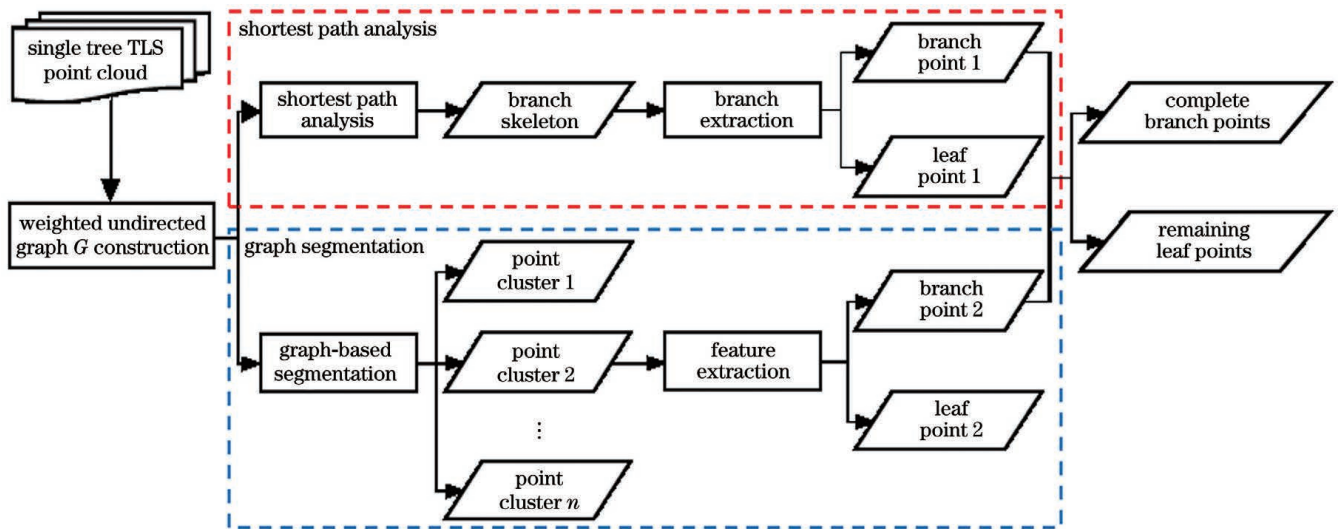


图 1 方法流程图

Fig. 1 Flow chart of method

### 2.1 最短路径分析

最短路径算法和图分割算法均是在无向图的基础上进行的,因此首先将树木点云转换为无向图。无向图<sup>[17]</sup>表示为  $G=(V,E)$ ,其中  $V=v_1, \dots, v_i, \dots, v_n$  代表一组节点, $n$  代表点云中点的个数,每个节点  $v_i \in V$  对应于点云中的一个点; $E$  是无向图中边的集合,其中的边  $e_{ij}$  连接相邻点。许多研究中将两点之间的距离作为边权重,从而构建加权无向图,然后在此基础上计算所有节点到基本节点(点云中的最低点)的最短权重路径。但对于树木点云,以点间距离作为边权重构建无向图并不完全适用。树木中钝角形状的枝干点能够很好地表明树木的生长方向,当以点间距离作为边权重构建无向图时,在提取最短路径构建树木枝干骨架时,在钝角处可能会出现漏点现象,导致构建的骨架线可能会过于僵硬,无法很好地表示树木的生长结构。如图 2(a)所示,由于 A、B 点间距离与 B、C 点间距离之和大于 A、C 点间距离,因此在计算最短路径时直接连接 A、C。为了克服该类现象,本文通过大量实验验证后,以点间距离的平方作为无向图中的边权重,使得计算最短路径时 A、B、C 能够依次相连,如图 2(b)所示。而锐角形状生长的枝干附近的漏点对之后提取枝干骨架表示树木生长姿态的影响很小,因此可以忽略不计。

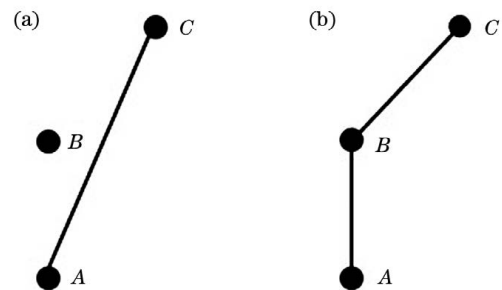


图 2 构建最短路径示意图。(a)以点间距离为权重构建最短路径;(b)以点间距离平方为权重构建最短路径

Fig. 2 Schematics of constructing shortest path. (a) Constructing shortest path with point spacing as weight; (b) constructing shortest path with square of point spacing as weight

无向图构建的具体步骤如下:

- 1) 输入树木点云  $Q$ ;
- 2) 确定  $Q$  中具有最小高程值的点  $d$  并创建空白图形  $G$ ;
- 3) 利用 KNN 算法搜索点云  $S$  中每个点周围  $n'$  个相邻点,存储其索引及相应点间欧氏距离的平方  $r^2$ ;
- 4) 将  $S$  中所有点加入图形  $G$  中,并从  $d$  开始,根据 KNN 算法搜索的结果,将  $r^2$  作为权重,向  $S$  中每

个点及步骤 3) 中搜索的  $n'$  个相邻点选择性添加加权边, 并设置权重阈值  $\delta$  判断两点是否相连;

5) 输出带权重的无向图  $G$ 。

利用单棵树木点云数据构建的无向图结构示意图如图 3(a) 所示。

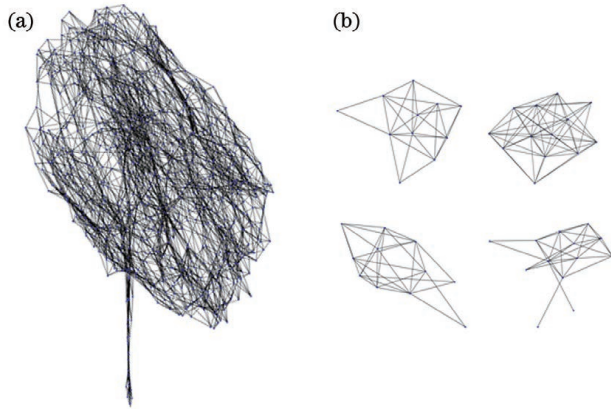


图 3 构建的无向图结构及分割后的图。(a) 构建的无向图; (b) 图分割结果

Fig. 3 Constructed undirected graph structure and trimmed graphs. (a) Constructed undirected graph; (b) result of graph segmentation

在完成无向图的构建后, 本文通过最短路径分析

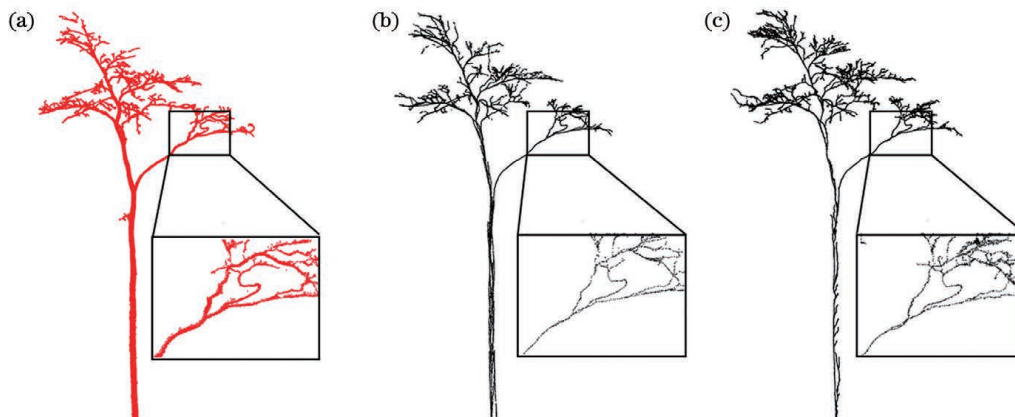


图 4 最短路径算法提取的枝干骨架。(a) 原始枝干点云; (b) 以点间距离为权重提取的枝干骨架; (c) 以点间距离的平方为权重提取的枝干骨架

Fig. 4 Skeletons of branches and trunks extracted by shortest path algorithm. (a) Original point cloud of branches and trunks; (b) skeleton of branches and trunks extracted with distance between points as weight; (c) skeleton of branches and trunks extracted with square of distance between points as weight

由图 4(b) 可知, 使用点间距离作为权重构建无向图将会使最短路径算法构建的枝干骨架过于僵硬, 无法较好地拟合原始树木中结构较弯曲的枝干, 而且在枝干分叉部分, 构建枝干分叉口时其位置与原始位置发生偏差。但以点间距离的平方作为权重构建无向图可以较好地解决这类问题, 提高骨架的柔软性, 为之后的枝干点提取打下基础。此外, 在相同阈值参数下, 以点间距离的平方为权重构建的枝干骨架更加完整。

通过该方法可以提取出树木点云中的细小分支结构, 其中, 树木末端枝干较细,  $k$  值不应设置过大, 否则会将细小枝干附近的树叶点错分为枝干点。但该方法

来检测出树干及树枝。首先, 利用 Dijkstra 算法计算出树木最低点到所有树木点的路径信息。在带权图中, Dijkstra 算法将单个节点固定为源节点, 找到从源节点到图中所有其他节点的最短路径, 从而生成整个树木点云的最短路径树。其次, 利用该最短路径树提取树木的枝干骨架<sup>[18]</sup>。由树木的生长结构可知, 在无向图中提取出的树木最低点到树叶点的最短权重路径必定是沿着树木枝干的, 因此与树叶中的节点相比, 树木主干以及分支中的节点在最短权重路径中出现的频率更大。树叶中的点通常对应最短路径中末端的部分节点, 因此在提取出的最短路径信息中, 将每条路径向底部回溯给定数量的步数  $R$ , 即去除路径末端  $R$  个点, 然后将所有路径信息进行汇总, 去除重复节点, 即可提取出树木的枝干骨架。最后, 在完成枝干骨架的提取后, 对枝干骨架中的每一个点执行 KNN 算法搜索, 将这个点周围距离最近的  $k$  个点设置为枝干点, 即可实现枝干点云的提取。其中  $k$  是基于树叶点与枝干点的密度选择的。

在利用单棵树木点云构建无向图的过程中, 本文分别使用了两点间的欧氏距离以及欧氏距离的平方对两点间的边进行加权, 经最短路径算法处理后, 所提取的枝干骨架如图 4(b)、(c) 所示。

对树木粗大枝干的提取效果不理想, 后续本文将通过结合图分割算法解决该问题。

## 2.2 图分割算法

枝叶点云分离的目的是为每个点赋予一个枝干或叶类标签。通过对点云进行稳健的分割, 充分地将点分割成簇, 每个点簇只包含纯叶点或纯木点, 再基于几何特征信息为各个点簇赋予标签, 即可实现树木的枝叶分离<sup>[19]</sup>。所以一个高效的点云分割方法可以大大提高枝叶分离的效率。

基于图的点云分割方法是将传统的区域生长方法转化为图连通分量搜索方法, 该方法也是在构建的无

向图上进行的,其目标是将图形分割成一块块区域,每一块区域对应一个点簇,点簇中的点具有相同特性,从而实现点云的分类。

图分割算法首先是将点云数据转换为无向图  $G$ , 具体转换方法已在第 2.1 节中介绍,其中每条边  $e_{ij} \in E$  都有一个对应的权重  $\omega_{ij}$ ,它是表示相邻元素  $v_i$  和  $v_j$  之间差异的非负度量,用于衡量由该边连接的两个点之间的差异,如空间位置、点云密度或其他一些局部的差异属性。然后将无向图分割成一组区域,使这些区域的并集对应于图中  $G'=(V, E')$ ,其中  $G'$  为分割后的无向图,  $E' \subseteq E$  是分割后的无向图中剩余边的集合。任何分割都是由  $E$  中边的子集引起的,利用实际应用中目标属性的差异,选择性去除集合中的部分边,将图形分割成多个区域。有不同衡量分割质量的方法,通常目标是使同一组件中的元素相似,不同组件中的元素不相似。因此,同一组件中两个顶点之间的边应该具有相对较低的权重,而不同组件中的顶点之间的边应该具有较高的权重。

基于第 2.1 节中构建的无向图,本文结合点云密度和逐点特征信息这两种特性,对图形中的边进行选择去除,从而将点云分割成簇,无向图的分割结果示意图如图 3(b)所示。其中,点云密度信息是指在 TLS 点云数据中,树木枝干点的密度通常大于树叶点的密度。逐点特征信息是每个点的垂直度,即法向量的  $z$  分量的绝对值<sup>[20]</sup>。

为了计算每个点的垂直度,首先要计算每个点的法向量。点的法向量是利用其相邻点构建的协方差矩阵来估计的,可视为协方差矩阵最小特征值的特征向量。对于点云中任意点向量  $\mathbf{P}_i$ ,其领域的协方差计算公式为

$$C_{\mathbf{P}_i} = \frac{\sum_{j=1}^{n'} (\mathbf{P}_j - \bar{\mathbf{P}}) (\mathbf{P}_j - \bar{\mathbf{P}})^T}{n'}, \quad (1)$$

$$\bar{\mathbf{P}} = \frac{\sum_{j=1}^{n'} \mathbf{P}_j}{n'}, \quad (2)$$

式中:  $\mathbf{P}_j$  为该邻域内  $j$  点的坐标向量,  $j=1, 2, \dots, n'$ ,  $n'$  是点向量  $\mathbf{P}_i$  领域内点的数量;  $\bar{\mathbf{P}}$  为该邻域的重心。

根据协方差矩阵可以求得三个特征值分别为  $\lambda_1$ 、 $\lambda_2$  和  $\lambda_3$ 。通常树木不同组成部分的点集具有不同的空间分布模式。其中,树叶点集是离散的,相应的三个特征值关系为  $\lambda_1 \approx \lambda_2 \approx \lambda_3$ ; 枝干点集近似于呈线性分布,相应的三个特征值关系为  $\lambda_1 \gg \lambda_2 \approx \lambda_3$ 。

在利用 KNN 算法构建的稠密连通图基础上,根据以下条件进一步修剪图形,将其分割成多个点簇,使每个点簇只包含纯枝干点或纯树叶点:

$$\begin{cases} |H_i - H_j| < H \\ d_{ij} < \bar{d}_i \end{cases}, \quad (3)$$

式中:  $H_i$  为  $i$  点的垂直度;  $H_j$  为  $j$  点的垂直度;  $H$  为设置的两点垂直度差值的阈值,用于确保连接点具有特征相似性;  $d_{ij}$  为  $i$  点到  $j$  点的欧氏距离;  $\bar{d}_i$  为  $i$  点到周围所有相邻点的平均距离,用于控制局部点云密度的连接性。

通过式(3)可以将无向图分割成多个点簇,其中只有具有相似特征的空间相邻点连接在一起。之后根据每个点簇的线性度  $L$  和点数  $S'$  来判断该点簇是否属于枝干点或树叶点。线性度  $L$  定义为

$$L = \frac{\lambda_1 - \lambda_2}{\lambda_1}. \quad (4)$$

本文在给定阈值范围内推断两个阈值所有组合情况下点的类别概率,计算点被分类为树叶点或枝干点的频率,通过比较分类为树叶点的频率和分类为枝干点的频率来判断点的类别。

最短路径算法与图分割算法都可以实现树木点云的枝叶分离,但两种算法的分离结果不同。如图 5 所示,

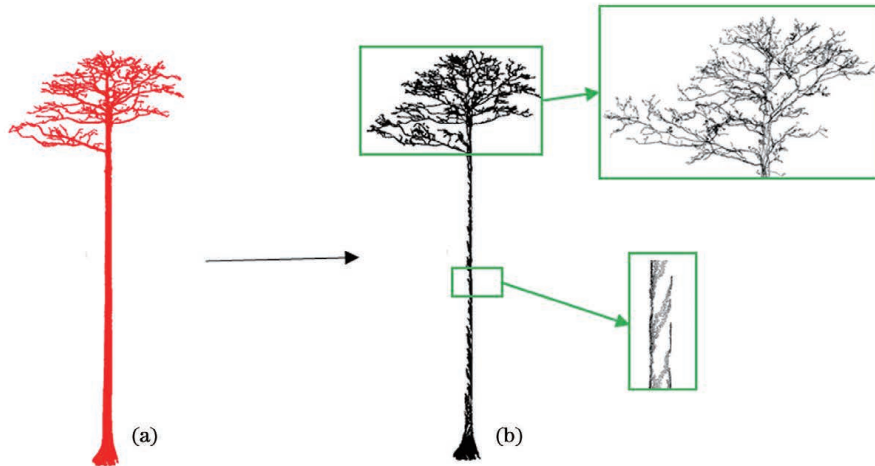


图 5 利用最短路径分析算法提取的树木枝干点云。(a)原始枝干点云;(b)提取的枝干点云

Fig. 5 Point cloud of tree branches and trunks extracted by shortest path analysis algorithm. (a) Original point cloud of branches and trunks; (b) extracted point cloud of branches and trunks

最短路径分析算法对细小枝干具有较好的分离结果，树木的较大枝干和主干的分离结果较差，产生漏提取现象。而如图 6 所示，图分割算法对树木的较大枝干

和主干的分离结果较好，对细小枝干的分离结果差。因此，本文在此基础上合并两种算法的提取结果，以实现高精度的树木点云枝叶分离。

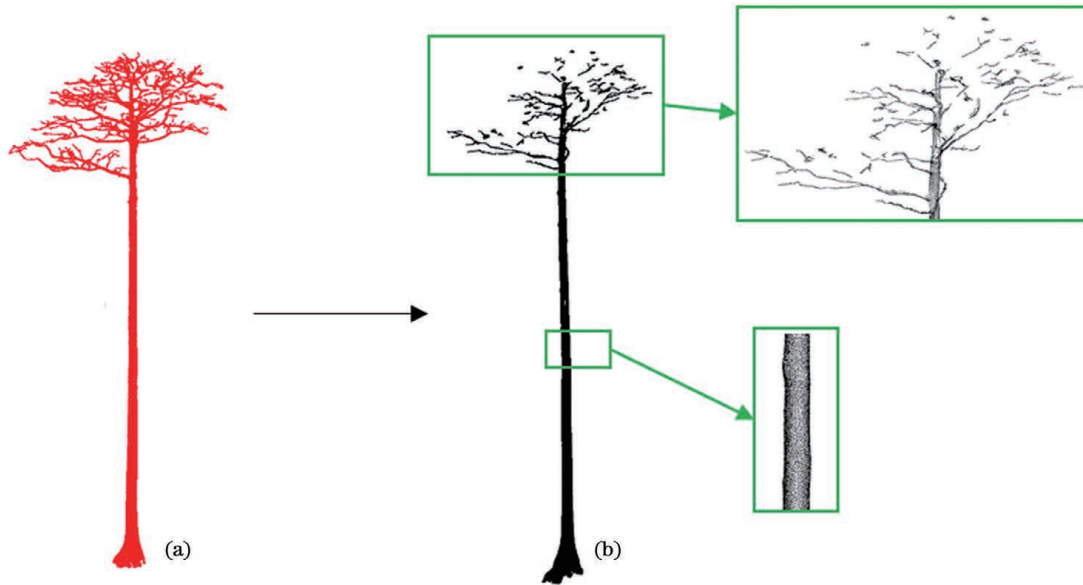


图 6 利用基于图的分割算法提取的树木枝干点云。(a)原始枝干点云;(b)提取的枝干点云

Fig. 6 Point cloud of tree branches and trunks extracted by graph-based segmentation algorithm. (a) Original point cloud of branches and trunks; (b) extracted point cloud of branches and trunks

### 2.3 分离结果合并

在合并过程中，由于部分点云被两种算法均识别为枝干点，因此在合并后需要去除其中的重复点，以避免数据冗余。而两种算法分离出的树叶点

可能均包含错分的枝干点，并不能直接进行合并，需要通过去除原始树木点云中最终被分类为枝干点的点云来进行提取。其分离结果合并流程如图 7 所示。

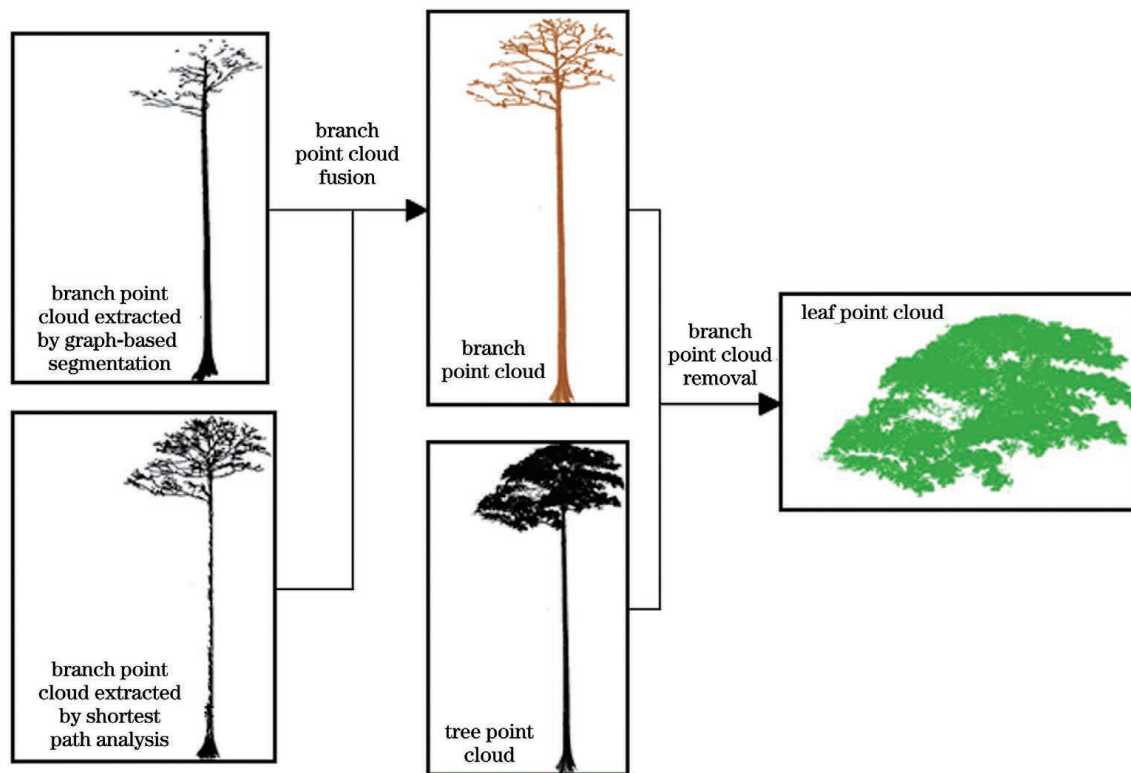


图 7 树木点云分离结果的合并流程图

Fig. 7 Merging flow chart of tree point cloud separation results

### 3 实验及结果

#### 3.1 评价指标

为了定量分析本文所提方法的枝叶分离结果,本文使用三个指标进行评估。 $A'$ 用来评价枝叶分离的精确度,其表达式为

$$A' = \frac{T_w + T_1}{T_w + F_w + T_1 + F_1}, \quad (5)$$

式中: $T_w$ 表示被正确分类为枝干点的点数; $F_w$ 表示被错误分类为枝干点的点数; $T_1$ 表示被正确分类为树叶点的点数; $F_1$ 表示被错误分类为树叶点的点数。

$F$ -score 是综合考虑精确率和召回率的调和值。其中,精确率表示被分类为枝干点的点云中分类正确的点所占的比例,计算方法为

$$P = \frac{T_w}{T_w + F_w}. \quad (6)$$

召回率表示在原始树木的枝干点云中,最后被正确分类为枝干的点所占的比例,计算方法为

$$R' = \frac{T_w}{T_w + F_1}. \quad (7)$$

$F$ -score 的计算式为

$$F = \frac{2 \times P \times R'}{P + R'}. \quad (8)$$

Kappa 值是用于评价分类结果一致性和信度的一个重要指标,可简称为  $\kappa$ ,其计算公式为

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \quad (9)$$

式中: $p_o$ 为比例一致性; $p_e$ 为随机一致性概率。计算公式分别为

$$p_e = \frac{(T_w + F_1) \times (T_w + F_w) + (F_w + T_1) \times (F_1 + T_1)}{(T_w + T_1 + F_w + F_1)^2}, \quad (10)$$

$$p_o = \frac{T_w + T_1}{T_w + T_1 + F_w + F_1}. \quad (11)$$

#### 3.2 实验数据

本文采用的实验数据为文献[8]中的开源数据,该数据集包含了多棵热带树木数据,大多具有扶壁结构,并且数据集提供了枝叶分离结果的参考值,其参考值是通过人工分离树木点云中的枝干点和树叶点获得的。由于树木结构的特殊性,对于真实采集的树木点云,无法获得其完整精确的枝干点和树叶点,因此相关研究中通常将人工分离结果作为标准结果。本文根据树木点云平均点间距的不同,在开源数据中选择三棵

树木进行实验研究,验证方法的稳定性,其数据描述和外观分别如表 1 和图 8 所示。

表 1 树木点云数据

Table 1 Tree point cloud data

Tree No.	Number of points	Tree height /m	Average point spacing /m
Tree 1	982636	52	0.12
Tree 2	947942	42	0.06
Tree 3	301615	36	0.04

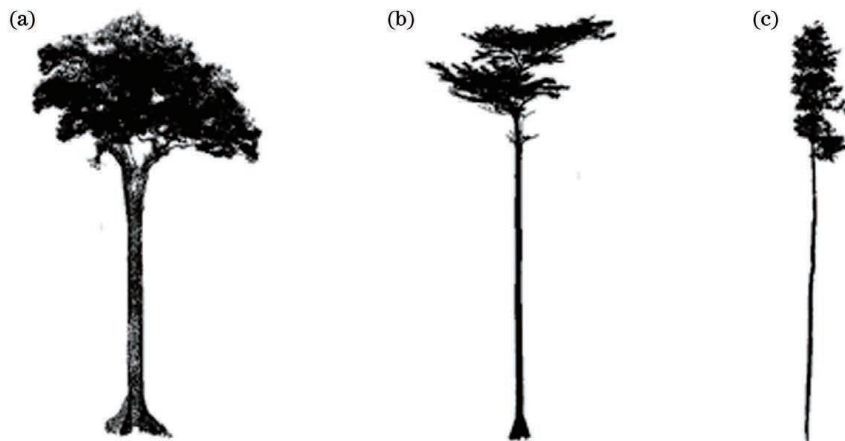


图 8 平均点间距不同的单棵树木点云。(a)Tree 1;(b)Tree 2;(c)Tree 3

Fig. 8 Single tree point clouds with different average point spacings. (a) Tree 1; (b) Tree 2; (c) Tree 3

此外,为了验证本文方法在不同数据质量和不同物种的树木点云数据下的分离能力,从开源数据集中选取了点云数据完整和缺失、树木有大果和小果或无果、树木有叶和叶少以及树种为阔叶和针叶的树木点云,两两相互对照,共 16 棵树木,其外观如图 9 所示。其中,

Tree 4 和 Tree 5 的树木点云数据较完整,Tree 6 和 Tree 7 的树木点云数据存在缺失,Tree 8 和 Tree 9 有大果,Tree 10 和 Tree 11 无果或有小果,Tree 12 和 Tree 13 有叶,Tree 14 和 Tree 15 叶少,Tree 16 和 Tree 17 为阔叶,Tree 18 和 Tree 19 为针叶。

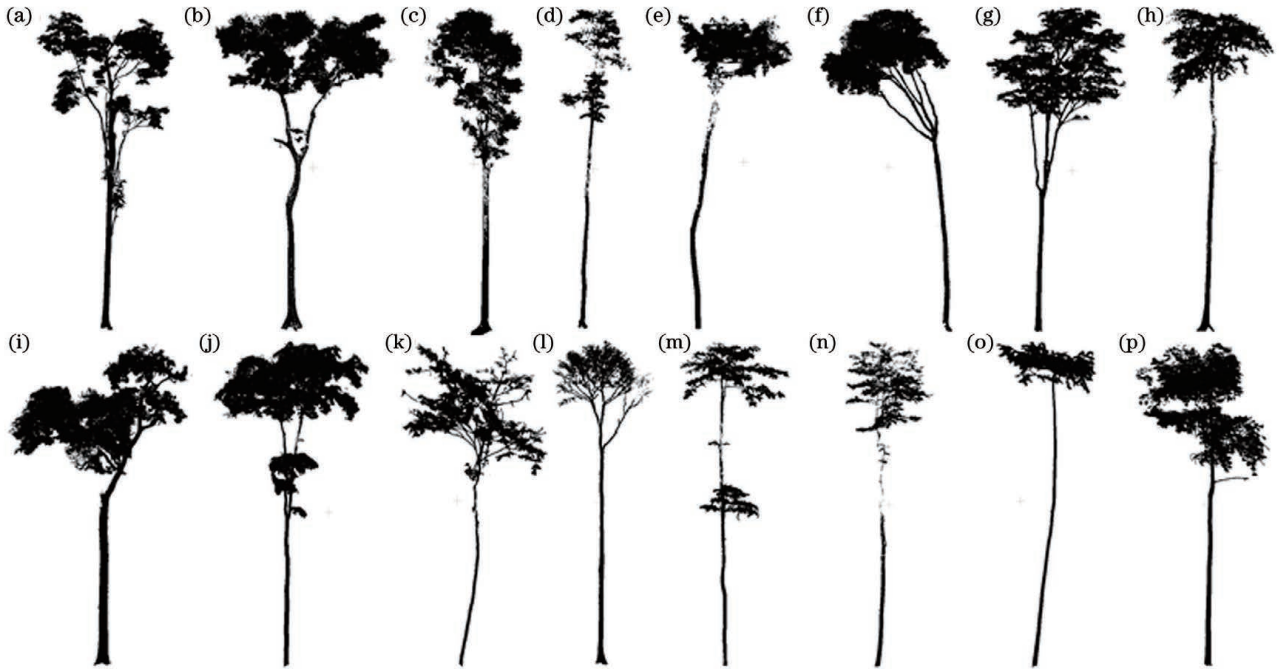


图 9 不同数据质量或树种的单棵树木点云。(a)Tree 4;(b)Tree 5;(c)Tree 6;(d)Tree 7;(e)Tree 8;(f)Tree 9;(g)Tree 10;(h)Tree 11;(i)Tree 12;(j)Tree 13;(k)Tree 14;(l)Tree 15;(m)Tree 16;(n)Tree 17;(o)Tree 18;(p)Tree 19

Fig. 9 Single-tree point clouds for different data quality or different tree species. (a) Tree 4; (b) Tree 5; (c) Tree 6; (d) Tree 7; (e) Tree 8; (f) Tree 9; (g) Tree 10; (h) Tree 11; (i) Tree 12; (j) Tree 13; (k) Tree 14; (l) Tree 15; (m) Tree 16; (n) Tree 17; (o) Tree 18; (p) Tree 19

### 3.3 实验结果

为了验证本文算法的枝叶分离能力,使用全部 19 棵树的实验数据进行研究,其中,使用伪随机的方法选择参数值对 Tree 1、Tree 2 及 Tree 3 三棵点间距不同的树木分别进行 20 次枝叶分离,每次测试和每棵树的参数值是伪随机选择的。可供随机选择的参数值如表 2 所示。路径回溯步数一般为 40~50,基于枝干骨架提取枝干点时  $k$  值一般为 10~20,而垂直度之差阈值一般为 0.08~0.11。另外,通过大量实验发现,线性度  $L$  阈值范围为 0.55~0.95,增量为 0.02;点数  $S'$  的阈值范围为 80~200,增量为 2,在这些参数下可以取得较好的枝叶分离结果。

表 2 用于验证测试的伪随机选择的输入参数值

Table 2 Pseudo-randomly selected input parameter values for verification testing

Parameter	Possible value
$R$	40,45,50
$k$	10,12,14,16,18,10,22,24,26,28
$H$	0.080,0.085,0.090,0.095,0.100,0.105,0.110
$L$	[0.55,0.95]
$S'$	[80,200]

使用实验数据进行枝叶分离,实验结果如表 3 所示,其中  $M$  为总体均值,SD 为标准差。每棵树的枝叶分离的平均准确度范围为 0.9264~0.9612,总体准确度的平均值为 0.9440,标准偏差为 0.0142。图 10 显示了每棵树进行 20 次枝叶分离的准确度。分析图 10 可知:Tree 1 的准确度最高,平均准确度为 0.9612,标准差为 0.0012,并且树叶分类结果也最好,平均  $F$  值为 0.9783,标准差为 0.0007;而 Tree 3 的分类结果呈现出最低的总体准确度和最大的标准差,且其他评价指标也相对较低。分析图 11 所示 Tree 3 的分离结果可知,本文方法在进行图分割时,由于该数据中部分主干点云过于稀疏,在分类时部分枝干点被错误地分类为树叶点。另外,文献[7]所提供的结果也是实验值而非标准值,可能存在错误。此外,每棵树的枝干点的  $F$  值大多低于树叶点的  $F$  值,Kappa 值一般均低于相

性度  $L$  阈值范围为 0.55~0.95,增量为 0.02;点数  $S'$  的阈值范围为 80~200,增量为 2,在这些参数下可以取得较好的枝叶分离结果。

表 3 每棵树的枝叶分离结果

Table 3 Separation result for branches and leaves of each tree

Tree No.	Accuracy		$F$ (wood)		$F$ (leaf)		$\kappa$	
	$M$	SD	$M$	SD	$M$	SD	$M$	SD
Tree 1	0.9612	0.0012	0.8454	0.0041	0.9783	0.0007	0.8183	0.0046
Tree 2	0.9445	0.0011	0.8278	0.0033	0.9504	0.0006	0.8223	0.0039
Tree 3	0.9264	0.0034	0.8024	0.0056	0.9547	0.0023	0.7572	0.0078

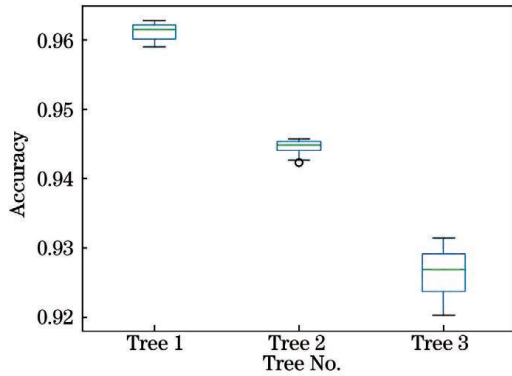


图 10 箱须图

Fig. 10 Box-and-whisker plot

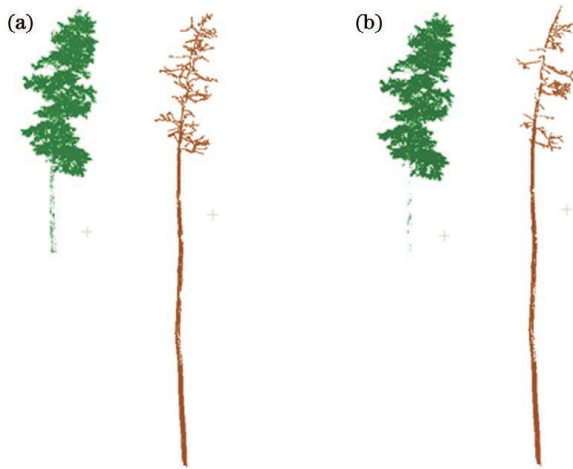


图 11 Tree 3 的枝叶分离结果及开源数据集中的参考值。  
(a)所提方法得到的枝叶分离结果;(b)开源数据集中的枝叶分离参考值

Fig. 11 Separation results for branches and leaves of Tree 3 and reference values in open source data set.  
(a) Separation result for branches and leaves by proposed method; (b) reference value for branch and leaf separation in open source data set

应的枝干点  $F$  值。总体而言,虽然通过伪随机的方法多次修改了输入参数,但每棵树每次枝叶分离的结果准确度均在 0.92 以上,较为完整地提取出了树木的枝干点,且各个评价指标的标准差均在 0.01 以下,表明本文所提出的方法具有较强的枝叶分离能力,对参数的敏感性不高,方法鲁棒性较好。

使用剩余 16 棵树木的点云数据进行枝叶分离,实验结果如表 4 所示。可以看出,所有树木的枝叶分离的精确度均在 0.9 以上,数据存在缺失的树木的枝叶分离结果的评价指标均低于数据完整的树木,但其精确度也在 0.93 以上,且二者树叶点的  $F$  值均高于枝干点的  $F$  值。而对于有果和无果的树木点云,其枝叶分离的结果相似,精确度均在 0.94 左右。利用本文方法在分离有叶和少叶树木点云时,可能是少叶的树木点云数据存在数据缺失且树木点云中存在散乱杂点,这使得少叶树木点云的枝叶分离结果相对不好。此

表 4 不同树的枝叶分离结果

Table 4 Separation results for branches and leaves from different trees

Tree No.	Accuracy	$F$ (wood)	$F$ (leaf)	$\kappa$
Tree 4	0.9689	0.9055	0.9814	0.8869
Tree 5	0.9854	0.8967	0.9921	0.8887
Tree 6	0.9431	0.8722	0.9631	0.8383
Tree 7	0.9315	0.8664	0.9522	0.8122
Tree 8	0.9589	0.8352	0.9765	0.8118
Tree 9	0.9309	0.7973	0.9538	0.7799
Tree 10	0.9335	0.8395	0.9581	0.7977
Tree 11	0.9519	0.8800	0.9699	0.8500
Tree 12	0.9548	0.8291	0.9739	0.8030
Tree 13	0.9861	0.8658	0.9814	0.8586
Tree 14	0.9342	0.8182	0.9598	0.7781
Tree 15	0.9346	0.9545	0.8831	0.8376
Tree 16	0.9491	0.8722	0.9682	0.8405
Tree 17	0.9773	0.9244	0.9867	0.9111
Tree 18	0.9641	0.8956	0.9784	0.8740
Tree 19	0.9090	0.7650	0.9525	0.7077

外,由于 Tree 15 中树叶点的数目少于枝干点的数目,树叶的  $F$  值低于枝干点的  $F$  值。对于阔叶和针叶树木,本文方法可以较好地分离阔叶树木,其分离精确度在 0.95 左右;而在处理针叶树木时,由于 Tree 19 的树叶点较多,且枝干较为杂乱,因此其枝叶分离结果的精确度不高,为 0.9090。由此可知,点云数据质量以及针叶和阔叶的不同对本文方法的枝叶分离能力有一定的影响,而树木果实的存在与否则影响不大。但总体而言,对于不同数据质量和不同树种的树木,本文方法均可以进行较好的枝叶分离,鲁棒性较高。

为了验证所提方法枝叶分离能力的优越性,选择 Tree 1、Tree 2 和 Tree 3 三棵具有代表性的树木以及 16 棵具有不同生长属性和数据质量的树木作为实验对象,对本文方法与 TLSeparation 方法<sup>[21]</sup>以及 LeWos 方法<sup>[19]</sup>进行比较。使用三种方法对 Tree 1、Tree 2 和 Tree 3 进行分类,结果分别如图 12~14 所示,19 棵树木的点云枝叶分离评价结果如表 5~7 所示。

分析表 5~7 可知,三种方法均可以在一定程度上实现单棵树木的枝叶分离,但相较于 TLSeparation 方法和 LeWos 方法,本文方法的各项评价指标均较高。特别是在面对小型树木、点云质量较低或细小枝干较多时,本文方法的评价指标明显高于 TLSeparation 方法,如 Tree 3、Tree 8 以及 Tree 15 等。TLSeparation 方法能够较为完整地分离出树木主干和较大枝干部分,但易于将枝干附近的树叶点分类为枝干点,如图 13 所示。这些细小的叶子会影响点的空间结构排



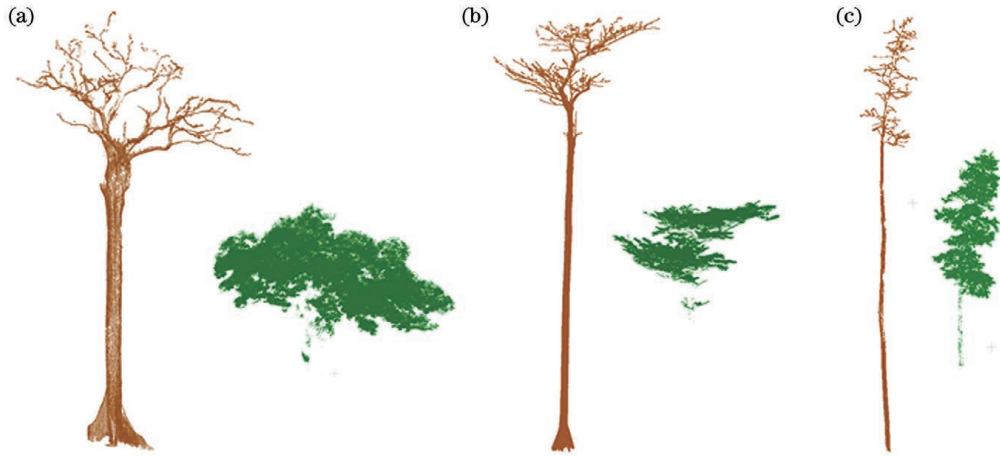


图 12 所提方法得到的枝叶分离结果。(a)Tree 1;(b)Tree 2;(c)Tree 3

Fig. 12 Separation results for branches and leaves by proposed method . (a) Tree 1; (b) Tree 2; (c) Tree 3

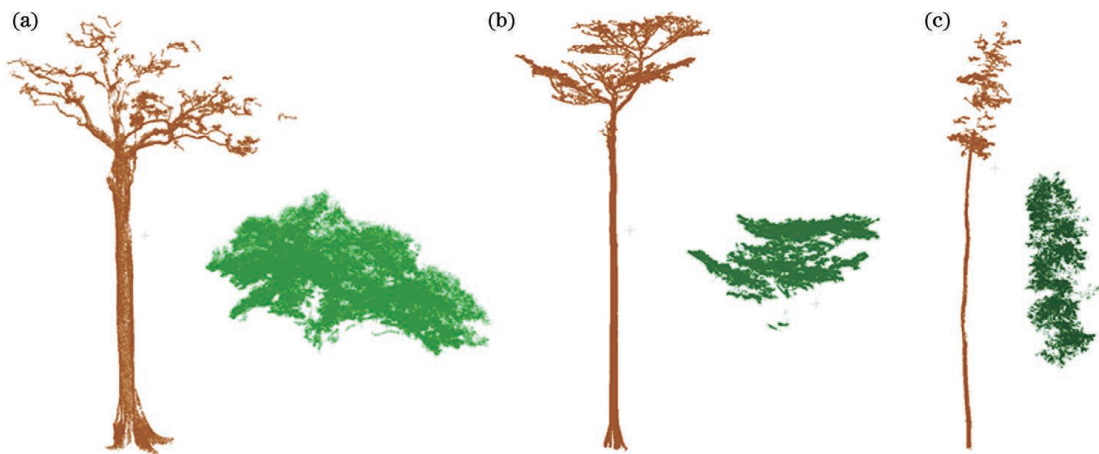


图 13 TLSeparation 方法得到的枝叶分离结果。(a)Tree 1;(b)Tree 2;(c)Tree 3

Fig. 13 Separation results for branches and leaves by TLSeparation method. (a) Tree 1; (b) Tree 2; (c) Tree 3

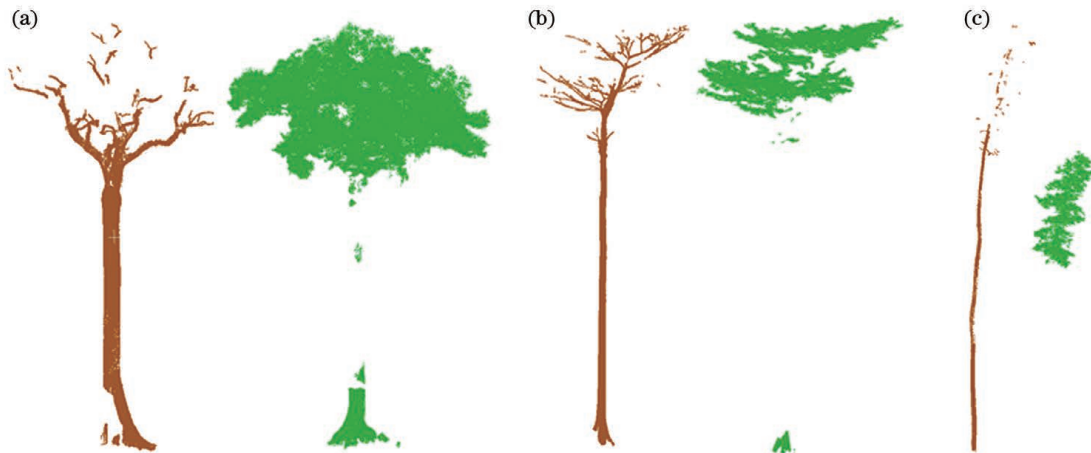


图 14 LeWos 方法得到的枝叶分离结果。(a)Tree 1;(b)Tree 2;(c)Tree 3

Fig. 14 Separation results for branches and leaves by LeWos method. (a) Tree 1; (b) Tree 2; (c) Tree 3

列,因此难以分离。但本文方法结合了最短路径分析算法,可以有效区分细小叶子并提取枝干,从而避免了树叶点的干扰。Lewos 方法在枝叶分离方面的评价指标较好,但面对树木的扶壁结构时容易出现错分现象,将部分扶壁结构的点云分类为树叶点,且无法有效分离树木的细小枝干,如图 14 所示。而本文方法通过

结合最短路径算法和图分割算法,可以较为完整地将主干提取出来。实验结果证明,本文所提方法具有较强的枝叶分离能力。

综上所述,在不需要过多人工干预的情况下,本文提出的算法可以以 90% 以上的准确率分离树木点云中的枝叶点。在 TLS 点云中分离枝干和树叶点云是

表 5 TLSeparation 方法在 19 棵树木的点云枝叶分离中的评价指标

Table 5 Evaluation indexes for TLSeparation method in separation of branches and leaves from 19 trees

Tree No.	Accuracy	$F(\text{wood})$	$F(\text{leaf})$	$\kappa$
Tree 1	0.9527	0.8035	0.9731	0.8338
Tree 2	0.8750	0.7780	0.9130	0.7690
Tree 3	0.6667	0.5131	0.7467	0.3327
Tree 4	0.9445	0.8218	0.9428	0.8046
Tree 5	0.9557	0.8000	0.9471	0.7875
Tree 6	0.9225	0.8136	0.9623	0.8062
Tree 7	0.9135	0.8832	0.9138	0.8271
Tree 8	0.8974	0.6500	0.9399	0.5918
Tree 9	0.9153	0.7472	0.9493	0.7186
Tree 10	0.9367	0.8530	0.9507	0.8127
Tree 11	0.9213	0.7769	0.9523	0.7303
Tree 12	0.9344	0.8148	0.9638	0.7986
Tree 13	0.9610	0.6877	0.9792	0.6681
Tree 14	0.8830	0.6493	0.9298	0.5792
Tree 15	0.8513	0.8855	0.7880	0.6788
Tree 16	0.9139	0.7967	0.9454	0.7424
Tree 17	0.9242	0.7232	0.9561	0.6799
Tree 18	0.9208	0.7840	0.9515	0.7359
Tree 19	0.9020	0.6949	0.9353	0.6527

表 6 LeWos 方法在 19 棵树木的点云枝叶分离中的评价指标  
Table 6 Evaluation indexes for LeWos method in separation of branches and leaves from 19 trees

Tree No.	Accuracy	$F(\text{wood})$	$F(\text{leaf})$	$\kappa$
Tree 1	0.9605	0.7899	0.9782	0.8405
Tree 2	0.9326	0.8645	0.9420	0.8337
Tree 3	0.9075	0.7902	0.9344	0.7365
Tree 4	0.9589	0.8818	0.9815	0.8634
Tree 5	0.9612	0.8611	0.9800	0.6087
Tree 6	0.9200	0.6263	0.9467	0.8080
Tree 7	0.9249	0.9052	0.9146	0.8898
Tree 8	0.9446	0.8287	0.9671	0.8020
Tree 9	0.9254	0.7069	0.9455	0.6756
Tree 10	0.9314	0.8391	0.9536	0.8133
Tree 11	0.9367	0.8319	0.9610	0.7932
Tree 12	0.9555	0.8582	0.9761	0.8644
Tree 13	0.9665	0.8526	0.9629	0.8456
Tree 14	0.9204	0.7650	0.9521	0.7171
Tree 15	0.9318	0.9532	0.8934	0.8867
Tree 16	0.9307	0.8301	0.9565	0.7866
Tree 17	0.9471	0.8315	0.9686	0.8002
Tree 18	0.9402	0.8214	0.9641	0.7855
Tree 19	0.9025	0.6634	0.9397	0.6155

表 7 本文方法在 19 棵树木的点云枝叶分离中的评价指标  
Table 7 Evaluation indexes for proposed method in separation of branches and leaves from 19 trees

Tree No.	Accuracy	$F(\text{wood})$	$F(\text{leaf})$	$\kappa$
Tree 1	0.9697	0.8674	0.9829	0.8475
Tree 2	0.9469	0.8863	0.9655	0.8547
Tree 3	0.9314	0.8110	0.9581	0.7691
Tree 4	0.9689	0.9055	0.9814	0.8869
Tree 5	0.9854	0.8967	0.9921	0.8887
Tree 6	0.9431	0.8722	0.9631	0.8383
Tree 7	0.9315	0.8664	0.9522	0.8122
Tree 8	0.9589	0.8352	0.9765	0.8118
Tree 9	0.9309	0.7973	0.9538	0.7799
Tree 10	0.9335	0.8395	0.9581	0.7977
Tree 11	0.9519	0.8800	0.9699	0.8500
Tree 12	0.9548	0.8291	0.9739	0.8030
Tree 13	0.9861	0.8658	0.9814	0.8586
Tree 14	0.9342	0.8182	0.9598	0.7781
Tree 15	0.9346	0.9545	0.8831	0.8376
Tree 16	0.9491	0.8722	0.9682	0.8405
Tree 17	0.9773	0.9244	0.9867	0.9111
Tree 18	0.9641	0.8956	0.9784	0.8740
Tree 19	0.9090	0.7650	0.9525	0.7077

单独估计枝干或树叶属性的关键步骤,本文所提方法可以有效进行高精度的树木点云枝叶分离,且方法鲁棒性较高。

## 4 结 论

提出了一种结合最短路径分析算法与图分割算法的树木点云枝叶分离方法,并通过具体实验验证了方法的可行性及鲁棒性。实验结果表明,该方法可以实现不同点间距树木的高精度枝叶分离,三类数据的分类精度分别达到了 0.9697、0.9469 和 0.9560, kappa 系数分别为 0.8475、0.8547 和 0.8925。所提方法得到的枝叶分离结果为后续的单木分析应用提供了参考。

但所提方法依然存在不足,在最短路径分析算法中的路径回溯部分,是对所有路径进行相同步数的回溯,但不同路径中末端树叶点的数目并不相同,在路径回溯时会将部分枝干加长或减短,降低了枝叶分离的准确度。此外,在基于枝干骨架提取枝干点时,  $k$  值的选取人工干预较大。未来将通过探索点云密度与  $k$  值的关系来实现参数的自适应选择,并根据路径中点的特征决定该路径的回溯步数,进一步提高算法的自动化程度以及分离能力。

## 参 考 文 献

- [1] 李春干, 代华兵. 中国森林资源调查: 历史、现状与趋势[J]. 世界林业研究, 2021, 34(6): 72-80.  
Li C G, Dai H B. Forest management inventory in China: history, current status and trend[J]. World Forestry Research, 2021, 34(6): 72-80.
- [2] 王斌. 基于地基激光雷达点云的树木三维结构自动重建技术研究[D]. 成都: 电子科技大学, 2015: 22-28.  
Wang B. Studies on automatic 3D reconstruction techniques of trees based on terrestrial lidar point clouds [D]. Chengdu: University of Electronic Science and Technology of China, 2015: 22-28.
- [3] 孙圆, 林秀云, 熊金鑫, 等. 基于地面激光强度校正数据的单木枝叶分离[J]. 中国激光, 2021, 48(1): 0104001.  
Sun Y, Lin X Y, Xiong J X, et al. Separation of single wood branches and leaves based on corrected TLS intensity data[J]. Chinese Journal of Lasers, 2021, 48(1): 0104001.
- [4] 肖杨, 胡少兴, 肖深, 等. 从三维激光点云中快速统计树木信息的方法[J]. 中国激光, 2018, 45(5): 0510007.  
Xiao Y, Hu S X, Xiao S, et al. A fast statistical method of tree information from 3D laser point clouds[J]. Chinese Journal of Lasers, 2018, 45(5): 0510007.
- [5] 梁增凯, 孙殿柱, 李延瑞, 等. 树状分层黎曼图约束的点云法向传播方法[J]. 光学学报, 2019, 39(12): 1211001.  
Liang Z K, Sun D Z, Li Y R, et al. Normal propagation of point clouds constrained by hierarchical riemannian graphs with tree structures[J]. Acta Optica Sinica, 2019, 39(12): 1211001.
- [6] 王嫛, 宋凯丽, 姚露露. 基于三维激光扫描的树木生物量碳储量测定[J]. 北京测绘, 2019, 33(10): 1200-1205.  
Wang Y, Song K L, Yao L L. Measurement of biomass and carbon storage of trees based on 3D laser scanning[J]. Beijing Surveying and Mapping, 2019, 33(10): 1200-1205.
- [7] 汪文琪, 李宗春, 付永健, 等. 基于改进多规则区域生长的点云多要素分割[J]. 光学学报, 2021, 41(5): 0528001.  
Wang W Q, Li Z C, Fu Y J, et al. Multi-factor segmentation of point cloud based on improved multi-rule region growing [J]. Acta Optica Sinica, 2021, 41(5): 0528001.
- [8] Momo Takoudjou S, Ploton P, Sonké B, et al. Using terrestrial laser scanning data to estimate large tropical trees biomass and calibrate allometric models: a comparison with traditional destructive approach [J]. Methods in Ecology and Evolution, 2018, 9(4): 905-916.
- [9] Li S H, Dai L Y, Wang H S, et al. Estimating leaf area density of individual trees using the point cloud segmentation of terrestrial LiDAR data and a voxel-based model [J]. Remote Sensing, 2017, 9(11): 1202.
- [10] Calders K, Newnham G, Burt A, et al. Nondestructive estimates of above-ground biomass using terrestrial laser scanning [J]. Methods in Ecology and Evolution, 2015, 6(2): 198-208.
- [11] 周薇, 马晓丹, 张丽娇, 等. 基于多源信息融合的果树冠层三维点云拼接方法研究[J]. 光学学报, 2014, 34(12): 1215003.  
Zhou W, Ma X D, Zhang L J, et al. Three dimensional point cloud splicing of tree canopy based on multi-source camera [J]. Acta Optica Sinica, 2014, 34(12): 1215003.
- [12] Beland M, Baldocchi D D, Widlowski J L, et al. On seeing the wood from the leaves and the role of voxel size in determining leaf area distribution of forests with terrestrial LiDAR [J]. Agricultural and Forest Meteorology, 2014, 184: 82-97.
- [13] Tao S L, Guo Q H, Su Y J, et al. A geometric method for wood-leaf separation using terrestrial and simulated lidar data [J]. Photogrammetric Engineering & Remote Sensing, 2015, 81(10): 767-776.
- [14] Ma L X, Zheng G, Eitel J U H, et al. Improved salient feature-based approach for automatically separating photosynthetic and nonphotosynthetic components within terrestrial lidar point cloud data of forest canopies [J]. IEEE Transactions on Geoscience and Remote Sensing, 2016, 54(2): 679-696.
- [15] Ferrara R, Virdis S G P, Ventura A, et al. An automated approach for wood-leaf separation from terrestrial LIDAR point clouds using the density based clustering algorithm DBSCAN [J]. Agricultural and Forest Meteorology, 2018, 262: 434-444.
- [16] Dijkstra E W. A note on two problems in connexion with graphs [J]. Numerische Mathematik, 1959, 1(1): 269-271.
- [17] Adams R, Bischof L. Seeded region growing [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994, 16(6): 641-647.
- [18] 李仁忠, 刘哲闻. 一种新的结合三维点云骨架点和特征点的分割方法[J]. 激光与光电子学进展, 2020, 57(12): 121104.  
Li R Z, Liu Z W. New segmentation method combining three-dimensional point cloud skeleton points and feature points [J]. Laser & Optoelectronics Progress, 2020, 57(12): 121104.
- [19] Wang D, Momo Takoudjou S, Casella E. LeWoS: a universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial LiDAR [J]. Methods in Ecology and Evolution, 2020, 11(3): 376-389.
- [20] Wang D, Brunner J, Ma Z Y, et al. Separating tree photosynthetic and non-photosynthetic components from point cloud data using dynamic segment merging [J]. Forests, 2018, 9(5): 252.
- [21] Vicari M B, Disney M, Wilkes P, et al. Leaf and wood classification framework for terrestrial LiDAR point clouds [J]. Methods in Ecology and Evolution, 2019, 10(5): 680-694.

## Tree Branch and Leaf Separation Using Terrestrial Laser Point Clouds

Lu Huaqing\*, Wu Jicang, Zhang Zijian

College of Surveying and Geo-Informatics, Tongji University, Shanghai 200092, China

## Abstract

**Objective** Terrestrial laser scanning (TLS) technology provides an efficient and accurate method for obtaining three-dimensional tree data. The separation of branches and leaves using single-tree point clouds is required when extracting tree structure parameters or determining the above-ground biomass. Additionally the separation of branches and leaves using TLS data enhances the ecological applicability of TLS data. Therefore, an efficient method for separating branches and leaves using tree point clouds can improve the application range of TLS. Existing branch and leaf separation methods either require precise calibration of lidar instruments to obtain intensity data for branch and leaf separation or use supervised classification methods, which require a considerable amount of manual intervention to select training data, and

retraining is required for trees from different environments or different tree species. They are not universal. To tackle these issues such as poor separation results, low separation efficiency, and low automation in the current TLS point cloud tree branch and leaf separation, this study further optimizes the unsupervised classification method based on geometric features and proposes a branch and leaf separation method combining the shortest path analysis algorithm and graph segmentation algorithm.

**Methods** The branch and leaf separation algorithm proposed in this study uses geometric features and structural analysis to classify the point cloud of a single tree into different components. First, a graph segmentation algorithm is used based on point geometric features and point cloud density. Subsequently, according to the three-dimensional coordinate vector  $P_i(x_i, y_i, z_i)$  of the point  $i$  in the point cloud, the algorithm constructs the covariance matrix of its neighborhood, calculates the three eigenvalues ( $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ ) of the corresponding point and the corresponding features vectors ( $e_1$ ,  $e_2$ , and  $e_3$ ) according to this matrix, and combines with the local point cloud density to carry out the coarse separation of branches and leaves. Second, it is the shortest path detection. The Dijkstra algorithm is used to calculate the shortest path from the lowest point in the tree point cloud to the remaining points. Then the skeleton of branches and trunks is extracted according to the growth structure of the tree, and the points of branches and trunks are extracted based on the skeleton of branches and trunks to realize the coarse separation of branches and leaves. Finally, the results of two coarse separations are combined to achieve the final fine separation of branches and leaves.

**Results and Discussions** This study employs three trees with different point spacings and 16 trees with different data quality from different tree species to perform quantitative and qualitative experiments to test the branch-leaf separation ability and robustness of the proposed method. First, three trees with different point spacings are separated 20 times using a pseudorandom method to select parameter values. Although the input parameters are modified several times by the pseudorandom method, the accuracy rate of each branch and leaf separation result of each tree is above 0.92. The branch points of the tree are extracted and the standard deviation of each evaluation index is observed to be below 0.01 (Table 3). Furthermore, using point cloud data of 16 trees with different data quality from different tree species for branch and leaf separation, the accuracy of branch and leaf separation of trees with missing data is not high, but the accuracies of branch and leaf separation of all trees are above 0.9 (Table 4). This indicates that the method proposed in this study has high branch and leaf separation ability and has good robustness. Moreover, the TLS separation method and the LeWos method are used to separate the branches and leaves of the three trees and the separation abilities are compared. The branch and leaf separation accuracies of the TLS separation method, the LeWos method, and the proposed method are compared (Tables 5–7). The classification indexes of this method are better than those of the TLS separation method and the LeWos method. Specifically, when the branches and leaves of the medium and small trees are separated, the evaluation indexes of the proposed method are significantly higher than those of the TLS separation method. Although the TLS separation method can thoroughly separate the tree trunk from the larger branches, it is easy to classify leaf points close to branches as branch points (Fig. 13). These tiny leaves affect the spatial structure of the dots, making it difficult to separate them. The evaluation indexes for the branch and leaf separation results of the LeWos method are good, but it is prone to misclassification when facing the buttress structure of trees. Some point clouds of some buttress structures are classified as leaf points and small branches of trees cannot be effectively separated (Fig. 14). The method proposed in this study, combined with the shortest path analysis algorithm, can effectively distinguish small leaves and extract branches to avoid the interference of leaves. The experimental results reveal that the proposed method has strong branch and leaf separation ability and high robustness.

**Conclusions** This study proposes a branch and leaf separation method that combines the shortest path analysis algorithm and the graph segmentation algorithm using tree point clouds. The feasibility and robustness of the proposed method are verified by specific experiments. The experimental results indicate that this method can realize high-precision branch and leaf separation for trees with different spacings. The classification accuracies on three types of data reach 0.9697, 0.9469 and 0.9560, respectively, and the kappa coefficients are 0.8475, 0.8547 and 0.8925, respectively. The branch and leaf separation results obtained by the method in this study serve as references for the subsequent application of single tree analysis.

**Key words** remote sensing; terrestrial laser scanning; branch and leaf separation; shortest path analysis; graph segmentation; point-by-point feature